Gesture Classification using Machine Learning with Advanced Boosting Methods

1st Abdurrahim Yilmaz
Yildiz Technical University
Mechatronics Engineering
Istanbul, Turkey
a.rahim.yilmaz@gmail.com

2nd Dilanur Bayraktar
Sisli Terakki Tepeoren High School
Istanbul, Turkey
dilanurbayraktar12@gmail.com

3rd Melih Akman
Yildiz Technical University
Mechatronics Engineering
Istanbul, Turkey
melihakman007@gmail.com

4th Cemre Sahinoglu
Yildiz Technical University
Mechatronics Engineering
Istanbul, Turkey
sahinoglucemre@gmail.com

5th Huseyin Uvet
Yildiz Technical University
Mechatronics Engineering
Istanbul, Turkey
huvet@yildiz.edu.tr

***Abstract*—In this paper, a detailed study on gesture classification using a dataset from Kaggle and optimizing the dataset is presented. The machine learning algorithms, which are SGD, kNN, SVM, MLP, Gaussian Naive Bayes classifier, Random Forest, LightGBM, XGBoost, and CatBoost classifiers, to conduct the research and, are used. The results are compared with each other to conclude which models perform the best in gesture classification. Except for the Gaussian Naive Bayes classifier, all methods resulted in high accuracy.**

***Index Terms*—Gesture classification, machine learning**

I. INTRODUCTION

Gesture classification using computer models has been an active research area for decades [1]. It is thought to be one of the most valuable research areas because it enables accesible communication and interaction between computers and humans. In fact, one of the examples for the applications of gesture classification is sign language translation [2]. Many researchers aspire to make human-computer interaction more convenient without the need for wearing any external device. The challenge for most researchers is that gesture classification requires high-resolution cameras to detect human body movements accurately. Plus, the visual data obtained - as photos or videos - can lead to inaccurate and imprecise results due to uncontrollable conditions like lighting, camera movement, action variability, etc. These are only some of the reasons why gesture classification models don't always turn out to be accurate, and researchers seek new ways of research in gesture classification. The most considerable improvement in gesture classification was when Microsoft Kinect was released. Being introduced to the market in 2010, the Kinect sensor uses Natural User Interface and, therefore, analyzed body movement, voice commands, gestures, facial expressions, etc. The Kinect sensor comprises a RGB camera, an IR (Infrared) depth sensor, an IR emitter, a microphone array for speech recognition, and a tilt motor to track body movements.

In this paragraph, the measures and approaches used by other researchers will be outlined. Paulo Trigueiros et al. [1] have applied the following learning algorithms on two different datasets: k-Nearest Neighbor (kNN), Naive Bayes (NB) classifier, Artificial Neural Network (ANN), and Support Vector Machines (SVM) [1]. They used Rapid Miner as a tool for this study. Also, in some parts that needed depth analysis, a Kinect camera was used. The results suggest that the ANN required more time to train the model but yielded the most accurate results. Another study carried out by Youness et al. [3] worked with Kinect sensors and applied SVM by using linear, polynomial, and radial basis function (RBF) kernels, ANN, kNN, and NB. Contrary to Trigueiros' study, Choubik et al. [3] found that SVM had the best performance in gesture classification. Bhattacharya et al. [2] also used Kinect, and they compared the accuracy of Decision Tree (DT) and SVM (linear and RBF kernel) algorithms on Kernel data. Although both DT and SVM proved to perform accurately, with DT having an accuracy of 99.32% and SVM having an accuracy of 99.97%, SVM was more effective in gesture learning. Later, linear kernel and RBF in SVM were compared in terms of their accuracy in gesture classification. The overall trend they found was that SVM (linear kernel) performed better, following with SVM (RBF kernel), and the least accurate being DT.

The significance of this paper is to show how gesture classification can be done by using a relevant dataset found on Kaggle and using Google Colab. This paper will explain, how we used Kinect to produce an accurate gesture classification model in detail. We will outline whether other approaches could also be used to yield better results. Also, it will be shown how the machine learning algorithms were optimized and improved to deliver the most accurate results for gesture classification. So, this paper will cast light on the methodology behind developing machine learning models with advanced boosting methods to get precise results in gesture classification.

II. Methodology

In this section, the crucial algorithms that can be used for the study, which are k-Nearest Neighbors (kNN), Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF), Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), and CatBoost will be presented in detail. The dataset that was utilized for this research was used to generate a classification model [4]. The dataset contains data about the joint of a person to determine the gesture. In this case, the gesture was classified into six different groups: Y, sumo, mermaid, seated, towel, wall [4].

A. Machine Learning

Machine learning algorithms are usually used for developing models through methods like regression or classification. Each algorithm uses different measurements to build the models. That's why each of them has distinct advantages and disadvantages. It wouldn't be wrong to state that different algorithms should be tested for a specific area of research to find the one that provides the most accuracy. This is because the unique measurements of different algorithms might be more applicable in some situations than others.

1) Stochastic Gradient Descent Classifier: SGD classifier that uses convex loss function is a type of gradient descent, but it can converge fastly than gradient descent. It can be implemented to large-scale data successfully [5].

2) K-Nearest Neighbor Classifier: kNN holds the assumption that similar data points are close to each other. This closeness can be thought of as proximity or distance between the points found through mathematical calculations. The advantages of kNN are that it is easy to implement, and there is no need for additional assumptions. However, a critical disadvantage of kNN is that the algorithm gets so slow as the number of variables increases. That's why kNN is not pragmatic for building models that contain over much data that need to be dealt with rapidly [6].

3) Support Vector Machine Classifier: The Support Vector Machine Classifier is used to classify data points by finding the most relevant hyperplane in an N-dimensional space. To yield the most accurate results, it is of great significance to find a hyperplane with the greatest margin, specifically the greatest distance between the classes [7].

4) Decision Tree Classifier: The Decision Tree Classifier is used for clearly organizing and classifying data. The decision tree is often made up of three parts: the nodes, the links, and the leaves. Nodes represent the features, the links represent the decisions, and the leaves represent the results. The objective is to minimize the errors in each leaf [8].

5) Random Forest Classifier: The Random Forest Classifier consists of multiple decision trees to obtain the best possible results. In a random forest algorithm, each decision tree predicts an outcome, and the prediction that gets the most votes becomes the overall model prediction. The key to getting accurate results in a random forest is to have decision trees whose results have low correlation values [9].

6) XGBoost Classifier: XGBoost (eXtreme Gradient Boosting) is a machine learning method that makes Gradient Boosting algorithms high-performance versions with several arrangements. The main motives reasons why most developers choose this method are that of obtaining high prediction, preventing over-learning, and managing open data in quickly. Tianqi's [10] studies show that, XGBoost works much faster than other well-known algorithms.

7) LightGBM Classifier: LightGBM (Light Gradient Boosting Machine) is a tree-based machine learning classification method originally developed by Microsoft. The significant properties of this classifier are faster training speed, higher efficiency, lower memory usage, better accuracy, and capability of large-scale handling data [11].

8) CatBoost Classifier: CatBoost Classifier is an open-source library developed by Yandex. It grants gradient boosting on decision trees. The significance of this algorithm is that it is faster than other Gradient boosting libraries.[12]

B. Metrics

In this section, an introduction of metrics is provided to make the paper more comprehensive, as shown in Table II-B. Metrics are used to make interpretations of the model's performance. The metrics can include various aspects, such as accuracy and precision that can be calculated through a confusion matrix. However, in this study, the following metrics are used to derive a conclusion of the models' performance in gesture classification: accuracy, precision, recall, and F_1 score. The metrics can be calculated by using the essential parameters according to the following criteria:

- True Positive (TP) occurs when the model's positive prediction is correct.
- True Negative (TN) occurs when the model's negative prediction is correct.
- False Positive (FP) occurs when the model's positive prediction is wrong.
- False Negative (FN) occurs when the model's negative prediction is wrong.

TABLE I
Metrics

Metric	Formula
Accuracy	$\frac{TP+TN}{TP+FP+TN+F}$
Recall	$\frac{TP}{TP+FN}$
Precision	$\frac{TP}{TP+FP}$
F_1 Score	$2 * \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

III. Results

The performances of each model to see which model yields the best result in hand gesture classification were obtained and will be presented in this section. Table II represents the

TABLE II
WEIGHTED RESULTS OF METRICS OF MACHINE LEARNING APPROACHES

Metrics \ Name	Result								
	SGD	kNN	SVM	MLP Classifier	Gaussian NB	Random Forest	XGBoost	LightGBM	CatBoost
Accuracy (%)	97	100	99	100	82	100	100	100	100
Recall (%)	97	100	99	100	82	100	100	100	100
Precision (%)	98	100	99	100	84	100	100	100	100
F_1 Score (%)	97	100	99	100	77	100	100	100	100

results concisely and transparently, but the outcomes of each metric used for determining the performance of the model will be explained in detail in this section. Firstly, SGD has an accuracy of 97%, recall of 97%, precision of 98%, and F_1 score of 97%. Because of this reason, while it wouldn't be correct to state that SGD is the best model for gesture classification, it could still be used to get results with an above-average performance. kNN has an accuracy, recall, precision, and F_1 score of 100%, which means that it is one of the best models for gesture classification. The SVM classifier's outcome was a 99% accuracy, recall, precision, and F_1 score. While SVM is better at performing hand gesture classification than SGD, it couldn't surpass the performance of kNN. Then, there is the MLP classifier with accuracy, recall, precision, and F_1 score of 100%. Results suggest that besides, kNN, MLP is also an excellent model to achieve viable results in gesture classification. The Gaussian classifier - with accuracy and recall of 82%, precision of 84% and F_1 score of 77% - is the model that performs the worst in gesture classification among all the models. Lastly, there are the Random Forest Classifier, LightGBM, CatBoost and XGBoost models that generated the same results as kNN and MLP Classifier, with accuracy, recall, precision, and F_1 score of 100%. So, the best performance was gained by the kNN, MLP, Random Forest Classifiers, XGBoost, LightGBM and CatBoost in gesture classification. The second best results were obtained through SVM, following with SGD, and the worst performance being the Gaussian Naive Bayes classifier.

IV. CONCLUSION

In this study, the aim was to find the machine learning algorithm that performs the best and achieves the most likely results in the well-known research area of gesture classification. Fortunately, the aim was met by selecting an appropriate dataset from Kaggle and optimizing the dataset with multiple models, including SGD, kNN, SVM, MLP, Gaussian, Random Forest, LightGBM, CatBoost, and XGBoost. After the dataset was optimized with each of these models, the results were gathered and compared according to the model's accuracy, precision, recall, and F_1 score. Although almost every model performed above-average can be used for gesture classification, the best results (100% accuracy, precision, recall, and F_1 score) were obtained by the kNN, MLP, Random Forest Classifiers, LightGBM, CatBoost and XGBoost. To sum up, the importance of this paper is to contribute to a general the research area of gesture classification by contrasting multiple model's performances and results.

REFERENCES

[1] P. Trigueiros, F. Ribeiro, and L. P. Reis, "A comparison of machine learning algorithms applied to hand gesture recognition," in *7th Iberian conference on information systems and technologies (CISTI 2012)*, pp. 1–6, IEEE, 2012.

[2] S. Bhattacharya, B. Czejdo, and N. Perez, "Gesture classification with machine learning using kinect sensor data," in *2012 Third International Conference on Emerging Applications of Information Technology*, pp. 348–351, IEEE, 2012.

[3] Y. Choubik and A. Mahmoudi, "Machine learning for real-time poses classification using kinect skeleton data," in *2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*, pp. 307–311, IEEE, 2016.

[4] B. Capella, D. Subramanian, R. Klatzky, and D. Siewiorek, "3d kinect total body database for back stretches," May 2019.

[5] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, Springer, 2010.

[6] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[7] C. Cortes and V. Vapnik, "Support vector machine," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[8] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[9] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[10] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, 2015.

[11] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.

[12] A. V. Dorogush, V. Ershov, and A. Gulin, "Catboost: gradient boosting with categorical features support," *arXiv preprint arXiv:1810.11363*, 2018.